Field Programmable Gate Arrays (FPGAs)

Logic cells imbedded in a general routing structure

- Logic cells usually contain:
  - N-input function calculator
  - Flip-flops

All features electronically (re)programmable
An FPGA

Note: Most pictures following courtesy of Mike Hutton/Altera Inc.

Altera Stratix II (EP2S60) Device Floorplan

60,440 Equivalent Logic Elements
2,544,192 Memory Bits
The k-Input LUT (e.g. k=4)

LUT-mask

\[ a'b'c'd' + abcd + abc'd' = 1000\,0000\,0000\,1001 = 0x8009 \]

Adaptive Logic Module
Stratix II ALM

LAB Interface

Input Output
Hierarchy: LAB / Cluster / CLB

Routing
Routing (Detailed)

Xilinx 4000 Series Routing Details
### TriMatrix Memory in Stratix Devices

<table>
<thead>
<tr>
<th>M512 Blocks</th>
<th>M4K Blocks</th>
<th>M-RAM</th>
<th>Dedicated External Memory Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small First-in First-out (FIFO) Buffers</td>
<td>Header / Cell Storage</td>
<td>Packet / Data Storage During Processing</td>
<td>Look-Up Schemes</td>
</tr>
<tr>
<td>Rate Changing</td>
<td>Channelized Functions</td>
<td>Program Memory for Nios® Embedded Processors</td>
<td>Packet &amp; Cell Buffering</td>
</tr>
<tr>
<td>Shift Register</td>
<td></td>
<td></td>
<td>Cache</td>
</tr>
<tr>
<td>Clock Domain Buffering</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **More Bits for Larger Memory Buffering**
- **More Data Ports for Greater Memory Bandwidth**

---

### Stratix DSP Blocks

- **18-Bit x 18-Bit Multiplier**
- **52-Bit Accumulator**
- **Input Register Unit**
- **Optional Pipelining**
- **Output Register Unit**
Putting it Together

FPGA Roles

- Digital logic implementation & prototyping
- Multi-mode systems
  - Change functionality for different applications
- Logic emulation
- Stream-based computing
- Processor acceleration

![Diagram showing FPGA processing of raw image data to processed image](image-url)
Partitioning

For Multi-FPGA System:
- Break logic into individual FPGAs
- Respect inter-FPGA communications
- Similar to placement

Techniques
- Multi-level partitioning (xbars)
- Simulated Annealing

Generally an unsolved problem

Virtual Wires

Multi-FPGA systems typically pin-limited, not logic limited
- FPGA: up to 1 Million logic gates, 512 I/Os.
- Partitioned circuit components might be:
  - 10x(1 Million gates, 5,000 I/Os)
  - 100x(100,000 gates, 500 I/Os)

Solution:
- 20x(1/2 Million gates, 2,500 I/Os + time division multiplexing on I/Os)
Global (Inter-FPGA) Routing

Route from source to destination FPGA using fixed resources
Similar to Aphyds Global Routing, but with fixed capacities
Maze, Steiner, etc. all can be applied
Must deal with potentially non-geometric distances

Technology Mapping

Take circuit and map it into the basic elements of the FPGA
5-LUTs
Must consider multiple factors
logic decomposition
logic replication
reconvergent fanout
Placement

Assign logic blocks to specific chip locations
Virtually identical to Aphyds Placement
Seek to minimize routing distance, congestion

FPGA Routing

Must pick the individual resources to use to carry a signal
- fixed capacity
- potentially non-geometric distances
- balance demands of multiple routes

Pathfinder (McMurchie, Ebeling)
- Convert routing architecture to graph
- Ignore congestion – change penalties and iterate
- Use maze + A* routing
- Integrate performance and congestion avoidance into one algorithm
**Pathfinder**

Represent all interconnection resources as a directed graph
Pin permutations on LUT inputs also captured

Routing sketch:
Each iteration rip-up and reroute all signals independently.
Resources currently used by another net cost more
Between iterations increase cost of resources that are shared

-> Over time, signals “bid” on preferred route, negotiating a compromise

---

**FPGAs & Multi-FPGA Systems**

Fit logic into a prefabricated system
Fixed inter-chip routing
Fixed on-chip logic & routing

---
### CAD & Physical Design

**CAD** = Computer Aided Design

Complexity of today's circuits requires computer support for most design tasks

CAD split into Synthesis, Physical Design

Synthesis = translating designer requirements into a circuit graph

PD = translating circuit graph into layout ("blueprint") for fabrication

<table>
<thead>
<tr>
<th>Processes</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Floorplanning</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Placement</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Global Routing</td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Detailed Routing</td>
<td><img src="image5.png" alt="Diagram" /></td>
</tr>
<tr>
<td>Compaction</td>
<td><img src="image6.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>